Report: Implementation of Tasks in WebGL Solar System Project

Introduction:

The purpose of this project is to implement a solar system simulation using WebGL, with a focus on scene graph management, lighting effects, and the addition of new celestial bodies.

Task 1: Scene Graph Transformation Implementation

In Task 1, the goal was to implement the draw function for the scene graph in the SceneNode class. This function is responsible for properly propagating transformations from parent to child nodes. The draw method was modified to apply transformations from the TRS object, consider the parent's model matrix if available, and recursively draw the children nodes. The successful implementation resulted in the expected output, displaying the Sun, Earth, and Moon in the solar system.

Task 2: Diffuse and Specular Lighting

Task 2 involved updating the fragment shader (meshFS) inside the MeshDrawer class to implement diffuse and specular lighting properly. The Lambertian reflection model was utilized for diffuse lighting, and the Phong reflection model was employed for specular lighting. The fragment shader now calculates both components based on the light direction, view direction, and reflection direction. The implementation successfully enhances the lighting effects, providing a more realistic representation of illuminated surfaces.

Task 3: Addition of Mars to the Solar System

In Task 3, the objective was to add Mars to the solar system. A new MeshDrawer and TRS were created for Mars, using a sphere as the mesh object. Mars was added as a child of the Sun in the scene graph, and specific transformations were applied: translation by -6 units on the X-axis, scaling to 0.35 for all coordinates, and a rotation of 1.5 times the Sun's rotation around its z-axis. The texture of Mars was sourced from an external image. The implementation successfully integrated Mars into the scene, providing a comprehensive representation of the solar system.

Conclusion:

The project achieved its objectives by implementing a WebGL-based solar system simulation. Task 1 ensured proper scene graph transformations, Task 2 enhanced lighting effects through the implementation of diffuse and specular lighting, and Task 3 successfully added Mars to the solar system with accurate transformations. The modular and object-oriented approach taken in the implementation allows for easy extension and maintenance of the codebase.

Future Enhancements:

Potential improvements for future development may include the addition of more celestial bodies, refining lighting models, and incorporating user interaction for a more interactive and engaging experience. Moreover, optimizations for performance and compatibility across different devices could be considered to enhance the project's accessibility.

Difficulties: However, it should not be seen a sun and moon when earth seen and earth cannot be seen when sun and moon appear. It is related to

```
sunTrs.setTranslation(0, 0, 5);
```

when this code change to as

```
sunTrs.setTranslation(0, 0, -5);
```

it is started to seen sun and moon but earth and mars dissapears.